

1. (Currently Amended): A method comprising:

calling a scheduling driver to start an Input/Output (I/O) request to a device for an
application, the device being one of a plurality of different types of devices useable by an
application;

determining if the device is busy; and

if the device is not busy,

providing an estimated processing time (EPT) for the I/O request to be completed for the
application, wherein the application sleeps for the estimated processing time.

2. (Original): The method of claim 1, wherein determining if the device is busy
comprises determining whether a locked flag is set, if the locked flag is set the device is busy and
if the locked flag is not set the device is not busy.

3. (Original): The method of claim 1, further comprising, setting a locked flag if the
device is not busy.

4. (Canceled)

5. (Currently Amended): The method of claim [4] 1, further comprising, calling the
scheduling driver to obtain I/O operation results after sleeping for the estimated processing time
and determining if the I/O request has been completed.

6. (Original): The method of claim 5, further comprising, clearing a locked flag if
the I/O request has been completed.

7. (Previously Amended): The method of claim 5, further comprising, providing the
I/O operation results from the I/O request if the I/O request has been completed.

1 8. (Original): The method of claim 5, further comprising, sleeping for a timer tick
2 interval if the I/O request has been completed.

1 9. (Previously Amended): The method of claim 5, further comprising, calculating
2 an estimated processing time remaining (EPTR) for the I/O request to be completed, if the I/O
3 request has not been completed, and providing the estimated processing time remaining (EPTR).

1 10. (Currently Amended): The method of claim 9, further comprising:
2 sleeping for the estimated processing time remaining (EPTR);
3 calling the scheduling driver to obtain the I/O operation results after sleeping for the
4 estimated processing time remaining (EPTR); and
5 determining if the I/O request has been completed.

1 11. (Currently Amended): The method of claim 10, further comprising:
2 determining if the I/O request has been completed and calculating an estimated
3 processing time remaining (EPTR) for the I/O request to be completed, if the I/O request has not
4 been completed;
5 sleeping for the estimated processing time remaining (EPTR);
6 calling the scheduling driver to obtain the I/O operation results after sleeping for the
7 estimated processing time remaining (EPTR); and
8 if the I/O request has not been completed,
9 [repetetively] repetitively performing the above operations until the I/O request has been
10 completed.

1 12. (Previously Amended): The method of claim 1, further comprising calculating an
2 estimated amount of time left (EATL) until the device will be available if the device is busy, and
3 providing the estimated amount of time left (EATL).

1 13. (Previously Amended): The method of claim 12, further comprising:
2 sleeping for the estimated amount of time left (EATL);
3 calling the scheduling driver to start the I/O request to the device after sleeping for the
4 estimated amount of time left (EATL); and
5 determining if the device is still busy.

1 14. (Currently Amended): The method of claim 13, further comprising:
2 determining if the device is still busy and calculating the estimated amount of time left
3 (EATL) until the device will be available, if the device is still busy;
4 sleeping for the estimated amount of time left (EATL);
5 calling the scheduling driver to start the I/O request to the device, after sleeping for the
6 estimated amount of time left (EATL); and
7 if the I/O request has not been started,
8 [repetetively] repetitively performing the above operations until the I/O request has been
9 started.

1 15. (Currently Amended): A machine-readable medium having stored thereon
2 instructions, which when executed by a machine, causes the machine to perform operations
3 comprising:
4 calling a scheduling driver to start an Input/Output (I/O) request to a device for an
5 application, the device being one of a plurality of different types of devices useable by an
6 application;
7 determining if the device is busy; and
8 if the device is not busy,
9 providing an estimated processing time (EPT) for the I/O request to be completed for the
10 application, wherein the application sleeps for the estimated processing time.

1 16. (Original): The machine-readable medium of claim 15, wherein determining if
2 the device is busy comprises determining whether a locked flag is set, if the locked flag is set the
3 device is busy and if the locked flag is not set the device is not busy.

1 17. (Original): The machine-readable medium of claim 15, further comprising the
2 operation of setting a locked flag if the device is not busy.

1 18. (Canceled)

1 19. (Currently Amended): The machine-readable medium of claim [18] 15, further
2 comprising the operations of calling the scheduling driver to obtain I/O operation results after
3 sleeping for the estimated processing time and determining if the I/O request has been
4 completed.

1 20. (Original): The machine-readable medium of claim 19, further comprising the
2 operation of clearing a locked flag if the I/O request has been completed.

1 21. (Previously Amended): The machine-readable medium of claim 19, further
2 comprising the operation of providing the I/O operation results from the I/O request if the I/O
3 request has been completed.

1 22. (Original): The machine-readable medium of claim 19, further comprising the
2 operation of sleeping for a timer tick interval if the I/O request has been completed.

1 23. (Previously Amended): The machine-readable medium of claim 19, further
2 comprising the operations of calculating an estimated processing time remaining (EPTR) for the
3 I/O request to be completed, if the I/O request has not been completed, and providing the
4 estimated processing time remaining (EPTR).

1 24. (Currently Amended): The machine-readable medium of claim 19, further
2 comprising the operations of:
3 sleeping for the estimated processing time remaining (EPTR);
4 calling the scheduling driver to obtain the I/O operation results after sleeping for the
5 estimated processing time remaining (EPTR); and
6 determining if the I/O request has been completed.

1 25. (Currently Amended): The machine-readable medium of claim 24, further
2 comprising performing the operations of:
3 determining if the I/O request has been completed and calculating an estimated
4 processing time remaining (EPTR) for the I/O request to be completed, if the I/O request has not
5 been completed;
6 sleeping for the estimated processing time remaining (EPTR);
7 calling the scheduling driver to obtain the I/O operation results after sleeping for the
8 estimated processing time remaining (EPTR); and
9 if the I/O request has not been completed,
10 [repetetively] repetitively performing the above operations until the I/O request has been
11 completed.

1 26. (Previously Amended): The machine-readable medium of claim 15, further
2 comprising the operations of calculating an estimated amount of time left (EATL) until the
3 device will be available if the device is busy, and providing the estimated amount of time left
4 (EATL).

1 27. (Previously Amended): The machine-readable medium of claim 26, further
2 comprising the operations of:
3 sleeping for the estimated amount of time left (EATL);

4 calling the scheduling driver to start the I/O request to the device after sleeping for the
5 estimated amount of time left (EATL); and
6 determining if the device is still busy.

28. (Currently Amended): The machine-readable medium of claim 27, further
comprising performing the operations of:

3 determining if the device is still busy and calculating the estimated amount of time left
4 (EATL) until the device will be available, if the device is still busy;
5 sleeping for the estimated amount of time left (EATL);
6 calling the scheduling driver to start the I/O request to the device, after sleeping for the
7 estimated amount of time left (EATL); and
8 if the I/O request has not been started,
9 [repetetively] repetitively performing the above operations until the I/O request has been
10 started.

29. (Currently Amended): An apparatus comprising:
1 a processor having a memory connected thereto, the memory storing an application, a
2 scheduling driver, the application calling the scheduling driver to start an Input/Output (I/O)
3 request to a device, the device being one of a plurality of different types of devices useable by an
4 application;
5 the scheduling driver,
6 determining if a device is busy; and
7 if the device is not busy,
8 providing an estimated processing time (EPT) for the I/O request to be completed
9 for the application, wherein the application sleeps for the estimated processing time.
10
11

1 30. (Original): The apparatus of claim 29, wherein determining if the device is busy
2 comprises determining whether a locked flag is set, if the locked flag is set the device is busy and
3 if the locked flag is not set the device is not busy.

1 31. (Original): The apparatus of claim 29, wherein the scheduling driver sets a locked
2 flag if the device is not busy.

1 32. (Canceled)

1 33. (Currently Amended): The apparatus of claim [32] 29, wherein the application
2 calls the scheduling driver to obtain I/O operation results after sleeping for the estimated
3 processing time and determines if the I/O request has been completed.

1 34. (Original): The apparatus of claim 33, wherein the scheduling driver clears a
2 locked flag if the I/O request has been completed.

1 35. (Original): The apparatus of claim 32 wherein the scheduling driver provides the
2 I/O operation results from the I/O request to the application if the I/O request has been
3 completed.

1 36. (Original): The apparatus of claim 32 wherein the application sleeps for a timer
2 tick interval if the I/O request has been completed.

1 37. (Original): The apparatus of claim 32 wherein the scheduling driver calculates an
2 estimated processing time remaining (EPTR) for the I/O request to be completed, if the I/O
3 request has not been completed, and provides the estimated processing time remaining (EPTR) to
4 the application.

1 38. (Currently Amended): The apparatus of claim 37, wherein the application:
2 sleeps for the estimated processing time remaining (EPTR);
3 calls the scheduling driver to obtain the I/O operation results after sleeping for the
4 estimated processing time remaining (EPTR); and
5 determines if the I/O request has been completed.

6 39. (Currently Amended): The apparatus of claim 38, wherein the application:
7 determines if the I/O request has been completed;
8 sleeps for the estimated processing time remaining (EPTR) calculated by the scheduling
9 driver;
10 calls the scheduling driver to obtain the I/O operation results after sleeping for the
11 estimated processing time remaining (EPTR); and
12 if the I/O request has not been completed,
13 [repetetively] repetitively performing the above operations until the I/O request has been
14 completed.

1 40. (Original): The apparatus of claim 29, wherein the scheduling driver calculates
2 an estimated amount of time left (EATL) until the device will be available to the application if
3 the device is busy, and provides the estimated amount of time left (EATL) to the application.

1 41. (Original): The apparatus of claim 40, wherein the application:
2 sleeps for the estimated amount of time left (EATL);
3 calls the scheduling driver to start the I/O request to the device for the application after
4 sleeping for the estimated amount of time left (EATL); and
5 determines if the device is still busy.

1 42. (Currently Amended): The apparatus of claim 41, wherein the application:

2
3
4
5
6
7
8

determines if the device is still busy;
sleeps for the estimated amount of time left (EATL) calculated by the scheduling driver;
calls the scheduling driver to start the I/O request to the device for the application, after
sleeping for the estimated amount of time left (EATL); and
if the I/O request has not been started,
[repetetively] repetitively performing the above operations until the I/O request has been
started.
